EXHIBIT 26

A Stepwise Algorithm for Finding Minimum Evolution Trees

Sudhir Kumar

Institute of Molecular Evolutionary Genetics and Department of Biology, The Pennsylvania State University

A stepwise algorithm for reconstructing minimum evolution (ME) trees from evolutionary distance data is proposed. In each step, a taxon that potentially has a neighbor (another taxon connected to it with a single interior node) is first chosen and then its true neighbor searched iteratively. For m taxa, at most $(m - 1)!/2$ trees are examined and the tree with the minimum sum of branch lengths (S) is chosen as the final tree. This algorithm provides simple strategies for restricting the tree space searched and allows us to implement efficient ways of dynamically computing the ordinary least squares estimates of S for the topologies examined. Using computer simulation, we found that the efficiency of the ME method in recovering the correct tree is similar to that of the neighbor-joining method (Saitou and Nei 1987). A more exhaustive search is unlikely to improve the efficiency of the ME method in finding the correct tree because the correct tree is almost always included in the tree space searched with this stepwise algorithm. The new algorithm finds trees for which S values may not be significantly different from that of the ME tree if the correct tree contains very small interior branches or if the pairwise distance estimates have large sampling errors. These topologies form a set of plausible alternatives to the ME tree and can be compared with each other using statistical tests based on the minimum evolution principle. The new algorithm makes it possible to use the ME method for large data sets.

Introduction

In the reconstruction of phylogenetic trees from a matrix of pairwise distances, the principle of minimum evolution (ME) is frequently used (Kidd and Sgaramella-Zonta 1971; Saitou and Nei 1987; Saitou and Imanishi 1989; Rzhetsky and Nei 1992, 1993). Rzhetsky and Nei (1993) have formally shown that the expected value of the sum of all branch lengths (S) is smallest for the true tree if an unbiased estimate of distances is used and the branch lengths are estimated by the ordinary least squares (OLS) method. An exhaustive search guarantees that the minimum evolution tree will be found, but it is not practical when many taxa (>10) are used. Consequently, several computationally less intensive algorithms have been proposed. For instance, the neighbor-joining (NJ) algorithm (Saitou and Nei 1987) combines a pair of sequences by minimizing the S value in each step of finding a pair of neighboring sequences. Because the S value is not minimized globally, the NJ tree may not be the ME tree if pairwise distances are not additive. Saitou and Imanishi (1989) showed that the NJ tree is very similar to the ME tree when the number of sequences used is small. This prompted various strategies of searching for the ME tree in the neighborhood of the NJ tree by conducting topological rearrangements (Rzhetsky and Nei 1992). However, these strategies may not work well when the number of sequences is large, especially when the correct tree contains many small interior branches (Rzhetsky and Nei 1992). It was then suggested that different tree topologies examined be generated by a bootstrap procedure (Rzhetsky and Nei 1994). However, the bootstrap method for generating alternative topologies can only be used if the original data can be resampled. In the bootstrap method much time is expended in resampling the data and re-estimating the pairwise distance matrix for generating topologies by the NJ method, which are not guaranteed to be distinct from the topologies found in previous replications. Moreover, the computation of OLS estimates of S independently for every topology examined may require a prohibitive amount of computer time.

In the following we present an algorithm that "heuristically" searches for the ME tree(s) in a stepwise manner similar to the NJ method and we suggest a dynamic procedure for efficiently computing the OLS estimates of S required to compare alternative topologies. We evaluate the efficiency of the new algorithm in recovering the correct tree by means of computer simulation. In this paper, we call the evolutionary entities (DNA sequences, species, etc.) taxa for the sake of convenience. Furthermore, a taxon is said to have a neighbor if it is connected to another taxon through a single interior node in the true tree (Saitou and Nei 1987). For instance, taxa 1, 2, 5, and 6 in figure 1*A* have neighbors, but the others do not. However, if we combine taxa 1 and 2 to form taxon 7, then 3, 7, 5, and 6 will have neighbors. Clearly, there are at least four taxa with neighbors in any bifurcating tree consisting of four or more taxa.

Algorithm

The proposed algorithm employs the fact that a bifurcating tree can be reconstructed in a stepwise fashion by first identifying a taxon (called leading taxon) that has a potential neighbor and then inferring its true neighbor. This algorithm is quite similar to the NJ method but, unlike with the latter, a large number of potential ME trees are examined and the misidentification of neighbors due to disturbing factors such as stochastic errors of nucleotide or amino acid substitutions is remedied to a large extent. Furthermore, all topologies examined by this algorithm will be different.

In the new algorithm, a leading taxon at the first step of search is first obtained. To accomplish this, we

Key words: phylogeny, minimum evolution trees, neighbor-joining, distance method.

Address for correspondence and reprints: Sudhir Kumar, 322 Mueller Laboratory, University Park, PA 16802. E-mail: imeg@psuvm.psu.edu.

Mol. Biol. Evol. 13(4):584–593. 1996

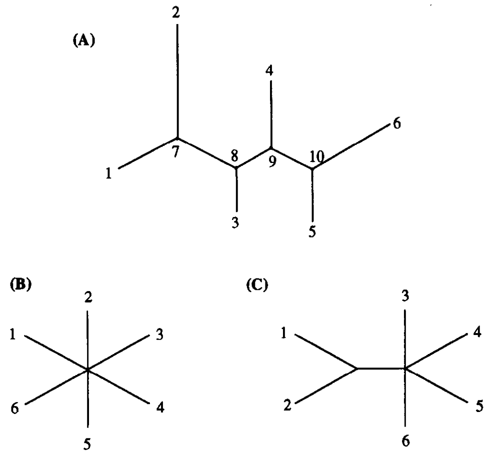

FIG. 1.—An unrooted tree of six taxa, 1–6. 7–10 are interior nodes. Modified from Saitou and Nei (1987).

begin with a star tree consisting of m taxa (fig. 1*B*) and compute the sum of its branch lengths (S_0) as

$$S_0 = \frac{1}{m-1}\sum_{i<j} d_{ij} = \frac{T}{m-1}, \tag{1}$$

where $T = \Sigma_{i<j}\, d_{ij}$. Now consider the tree in figure 1*C*, where taxa 1 and 2 have been paired. In this case, the sum of branch lengths (S_{12}) is given by

$$S_{12} = \frac{T}{m-2} - \frac{R_1 + R_2}{2(m-2)} + \frac{d_{12}}{2}, \tag{2}$$

where d_{12} represents the pairwise distance between taxa 1 and 2, and

$$R_1 = \sum_x d_{1x}, \qquad R_2 = \sum_x d_{2x} \tag{3}$$

(Saitou and Nei 1987; Studier and Keppler 1988; Nei 1990). In this fashion, we compute S_{ij} for all $m(m-1)/2$ possible pairs of taxa. Since the quantity $T/(m-2)$ is a constant in all S_{ij}'s, we need not compute it for comparing S_{ij} values.

Now we select a pair of taxa that gives the smallest S_{ij} value, say S_{ab}. For a purely additive tree, a and b are true neighbors, and either of them can be chosen as a leading taxon. However, the additivity condition is rarely satisfied with actual data, and a and b may not be true neighbors. Therefore, we find the smallest S from the two sets $\{S_{ai};\, i \neq b\}$ and $\{S_{bj};\, j \neq a\}$. If the minimum S_{ij} is found in the first set, a is chosen as the leading taxon; otherwise b is chosen (see Appendix). We now have a leading taxon for the first step of the taxon pairing.

Next, we arrange the $m - 1$ taxa to be paired with this leading taxon in the ascending order of their S_{ij} values (*i.e.*, from the best to the worst). We temporarily regard the first taxon in the list to be the neighbor of the leading taxon and combine them into a single composite taxon, and keep other potential neighbors for later use. The pairwise distance between the composite taxon u of the selected pair i and j and the other taxa (k; $k \neq i, j$) is then computed by the following equation (Studier and Keppler 1988)

$$d_{ku} = (d_{ik} + d_{jk} - d_{ij})/2. \tag{4}$$

From the resultant set of $m - 1$ taxa, we again select a leading taxon and pair it with one of the $m - 2$ remaining taxa in a manner similar to that of the first step. As before, we keep all other potential neighbors of this leading taxon for later use. We repeat this procedure until a leading taxon is selected at every step of taxon pairing ($m - 3$ steps). At this stage, we obtain the first tree, which is identical with the NJ tree, and compute its S value (S_{NJ}). Because we keep other potential neighbors of the leading taxon at each step of taxon pairing, we can go back and try each of them to search for a tree with the smallest S in a recursive manner (fig. 2).

In this algorithm, the choice of a correct leading taxon at every step of taxon pairing, except for the first step, is conditional on the selection of leading taxa and their potential neighbors in the previous steps. If correct leading taxa are selected in any one pass of the search, the set of topologies examined is guaranteed to include the correct tree. This is because all $n - 1$ possible pairs are tried in each step, where n is the number of unpaired taxa. It follows that at most $(m-1) \cdots 5 \cdot 4 \cdot 3 = (m-1)!/2$ distinct topologies (search paths) will be examined in this search (called the full search).

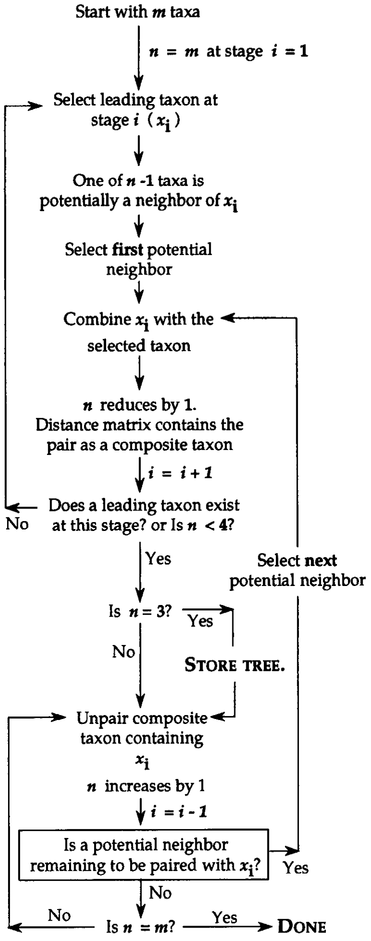
Start with m taxa
$n = m$ at stage $i = 1$
Select leading taxon at stage i (x_i)
One of n -1 taxa is potentially a neighbor of x_i
Select first potential neighbor
Combine x_i with the selected taxon
n reduces by 1. Distance matrix contains the pair as a composite taxon
$i = i + 1$
Does a leading taxon exist at this stage? or Is $n < 4$?
No
Yes
Select next potential neighbor
Is $n = 3$?
Yes
STORE TREE.
No
Unpair composite taxon containing x_i
n increases by 1
$i = i - 1$
Is a potential neighbor remaining to be paired with x_i?
Yes
No
No
Is $n = m$?
Yes
DONE

FIG. 2.—The flowchart of the Stepwise Minimum Evolution Tree algorithm. n refers to the number of unpaired taxa at a given stage of taxon pairing.

Table 1
Pairwise Distances and the S_{ij} Matrices for the Model Tree in Figure 3*A*

A. Pairwise distances

	1	2	*3*	*4*
2 . . .	1.3498			
3 . . .	1.6243	1.2696		
4 . . .	1.8943	1.6600	1.5576	
5 . . .	1.3744	1.5737	1.5263	1.4964

B. S_{ij} at the first step

	1	2	*3*	*4*
2 . . .	3.7677			
3 . . .	3.8842	3.7718		
4 . . .	3.9141	3.8619	3.7899	
5 . . .	3.7604	3.9250	3.8805	3.7605

C. S_{ij} at the second step[a]

i) When 5 and 1 paired to form 6

	6	2	3
2 . . .	2.3725		
3 . . .	2.4265	2.3589	
4 . . .	2.3589	2.4265	2.3725

ii) When 5 and 4 paired to form 6

	1	2	*3*
2 . . .	2.2340		
3 . . .	2.3214	2.2371	
6 . . .	2.2371	2.3214	2.2340

NOTE.—Constant factor $T/(m - 2)$ given in equation (2) was not included in the computation of S_{ij}.

[a] In case of ties, the pair first encountered was chosen. This choice does not affect the computation in the four-taxon case.

Below we give a simple example to illustrate how the new algorithm works in a stepwise manner: Consider the distance matrix given in table 1A, for which the tree in figure 3*A* is the correct tree. This distance matrix was obtained in one replication of the computer simulation (described later). In table 1B, the S_{ij} values computed by using equation (2) are given. In this table, the pair (1, 5) has the smallest S_{ij}, and S_{45} is smallest in the set $\{S_{12}, S_{13}, S_{14}, S_{52}, S_{53}, S_{54}\}$. Therefore, we select taxon 5 as the leading taxon and arrange the potential neighbors of taxon 5 in the ascending order of their S_{ij} values. Since taxon 1 heads this list, we combine (1, 5) to form a composite taxon 6 (fig. 3*C*). We now compute the S_{ij} for taxa 2, 3, 4, and 6 (table 1C[i]). We find that S_{46} is the smallest S_{ij} and that S_{26} is smallest in the set $\{S_{24}, S_{34}, S_{26}, S_{36}\}$, so taxon 6 is chosen as the leading taxon at this stage. We combine 6 and 4 into a composite taxon (called taxon 7), because taxon 4 is first in the list of potential neighbors. At this stage, there are only three unclustered taxa (2, 3, and 7) that can be connected only in one way. This is in fact the NJ tree but is different from the correct tree (fig. 3*A*). In any case, this is a temporary ME tree and the search proceeds further.

In figure 3*C*[i], we clustered the leading taxon 6 with 4, but there are two other potential neighbors of taxon 6, namely 2 and 3. Pairing 6 with 2, we obtained the tree in figure 3*C*[ii], and pairing 6 with 3 we obtained the tree in figure 3*C*[iii]. Both of these trees have a higher *S* value than the NJ tree.

Until now, we have examined all trees that were generated by the choice of taxon 1 as a potential neighbor for leading taxon 5 in the first step of the taxon pairing (fig. 3*B* and *C*). Next, we combine taxa 5 and 4 into a composite taxon 6 (fig. 3*D*). From the set of taxa 1, 2, 3, and 6 we again select a leading taxon (taxon 1 in this case; table 1C[ii]) and rank its potential neighbors in ascending order of their S_{ij} values. We pair taxa 1 and 2 and obtain the tree in figure 3*D*[i]. This tree has a smaller *S* than the NJ tree, so it becomes our new temporary ME tree. This tree is also identical with the true tree. Furthermore, the pairs (1, 6) and (1, 3) result in trees in figure 3*D*[ii] and [iii] that are more similar to the true tree than the NJ tree in terms of topology.

In a similar fashion, we combine taxon 5 with 3 and examine all possible resultant trees. We also combine taxon 5 with 2 and examine all possible trees (not shown). At the end, we would have examined 12 trees and found that the tree in figure 3*D*[i] is the ME tree.

Computationally Efficient Search Strategies

Strategies to Restrict the Tree Space Searched

As mentioned earlier, the full search using the stepwise algorithm will examine $(m - 1)!/2$ distinct topologies. Even though this number is considerably smaller than the number of all possible topologies (table 2), it increases rapidly with increasing number of taxa in the data set. There are many simple ways to reject unlikely topologies without examining them. Two of them are as follows.

First, if the S_{ij} for pairing the leading taxon with one of its potential neighbors is considerably larger than the smallest S_{ij} value in the search step (say, $S_{ij,s}$), this pair is unlikely to lead to a tree with a smaller *S*. Therefore, we may ignore all the pairs that show S_{ij} higher than $(1 + p)S_{ij,s}$. We call *p* a proportional search factor. In the following, we show the results of computer simulations with and without using this search strategy. (The values of S_{ij} and $S_{ij,s}$ do not include the constant $T/(m - 2)$ as mentioned earlier.) It is also possible to ignore the pairs for which $S_{ij} > (S_{ij,\max} - S_{ij,\min})p$, but this and other similar possibilities are not pursued in this paper.

Second, if some of the pairwise distances between a combined taxon and the other taxa become negative, we may ignore the search paths that are generated from these taxon pairs, since the actual branch lengths are not very likely to be less than 0. Of course, negative pairwise distances may occur even for correct taxon pairs if the corresponding branches are very short or if the estimates of distances have large sampling errors. In this case, in place of 0, we may use a small negative value as a threshold cutoff.

Fast Method for Computing Ordinary Least Squares Estimates of the *S* Value

When searching for the ME tree, the computation of the OLS estimates of *S* values for each tree independently would require a large amount of computational time and this strategy is not practical for even moder-

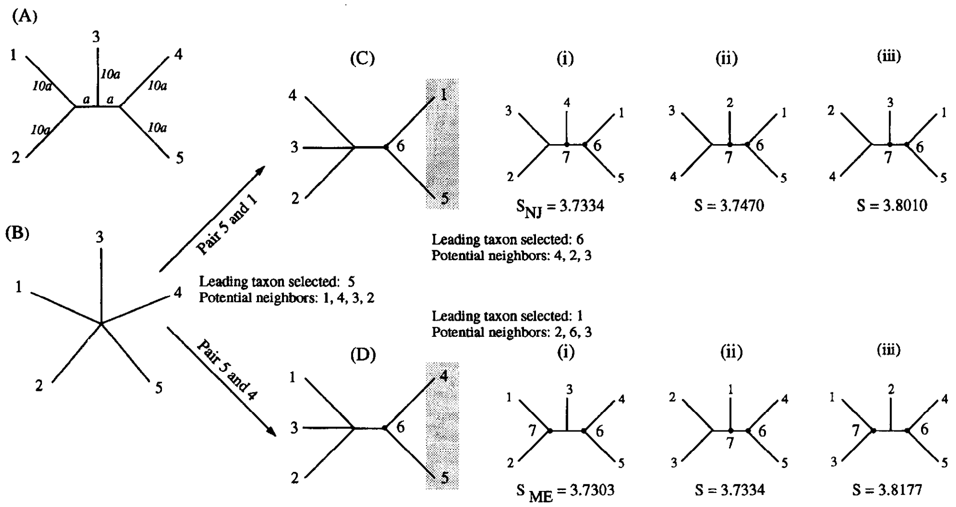

FIG. 3.—Application of the new algorithm to the distance matrix in table 1A that was generated by computer simulation using the tree in *A*; *a* = 0.06, nucleotide sequence length = 500. *S* is the sum of branch lengths. Only two of four paths leading from *B* are shown.

ately large data sets ($m > 10$). For algorithms in which the successive trees examined are different from each other in only a few branch rearrangements (e.g., the stepwise algorithm and Rzhetsky and Nei [1992] algorithm), the *S* values can be computed efficiently by noting that the OLS estimate of any interior (or exterior) branch length of a tree depends only on (1) the numbers of taxa in the four (or three) clusters of taxa connected to this branch and (2) the intercluster distances among these four (or three) groups (Rzhetsky, Kumar, and Nei 1995). So, as long as the taxa in each cluster remain the same in a given topology (the branching orders of taxa within each cluster are free to change), the branch length need not be re-estimated. By implementing a strategy that examines only the branches that have been disturbed from the time when the OLS estimates were last computed, the time required to search for the ME tree using OLS estimates of *S* is improved tremendously (fig. 4; compare OLS and FastOLS curves).

ME Criterion Based on a Simple Method of Estimating Branch Lengths

Theoretically, the *S* values obtained by using the OLS method are required for comparing different tree topologies (Rzhetsky and Nei 1993). However, in the present algorithm, the computation of the OLS estimate of a branch length requires additional $O(m^2)$ operations as compared to the Fitch and Margoliash (FM; 1967) method used for estimating branch lengths in the NJ method, which requires only O(1) more operations (see equation 6 in Saitou and Nei 1987). Thus the search for the ME tree can be speeded up considerably by using the FM method of estimating branch lengths (fig. 4; compare FM to OLS and FastOLS). However, the effect of this approach on the efficiency of the ME method in recovering the correct tree needs to be examined in the computer simulation.

Table 2
Number of Different Topologies Needed to be Examined

No. of Taxa	N_E	N_N	N_A
4	3	3	—
5	15	12	—
6	105	60	20
12	$10^{8.8}$	$10^{7.3}$	~350
100	10^{182}	10^{155}	—

NOTE.—N_E is the number needed for exhaustive search, N_N is the maximum number with the new algorithm, and N_A is the average number needed to be examined in the computer simulation results reported in this paper.

Computer Simulation

We considered four basic model trees: two constant-rate and two varying rates (among lineages) trees, each consisting of six taxa (fig. 5*A–D*; Saitou and Imanishi 1989). We then derived six composite 12-taxon trees: four consisting of two copies of the same tree (fig. 5*AA–DD*) and two consisting of one copy each of the two different trees in the same rate class (fig. 5*AB, CD*). Nucleotide sequences of length 300, 600, and 1,200 nucleotides and overall maximum pairwise divergences of about 0.1 (low) and 1.0 (high) substitutions per site were considered in the computer simulation.

The scheme of computer simulation used for all model trees is as follows. First, an ancestral sequence of a given number of nucleotides was generated with the assumption that all four nucleotides occur in equal frequency. This sequence was assumed to evolve according to a predetermined branching pattern of the model tree. The descendent sequence for a given branch was obtained by introducing random nucleotide substitutions in its immediate ancestral sequence. These random nucleotide substitutions were introduced following a Poisson distribution with the mean equal to the expected branch length. In this way, we obtained the sequence data for all terminal taxa in the model tree. The pairwise distances for the sets of sequences generated in this manner were then obtained (Jukes and Cantor 1969). This procedure was repeated to generate 500 rep-

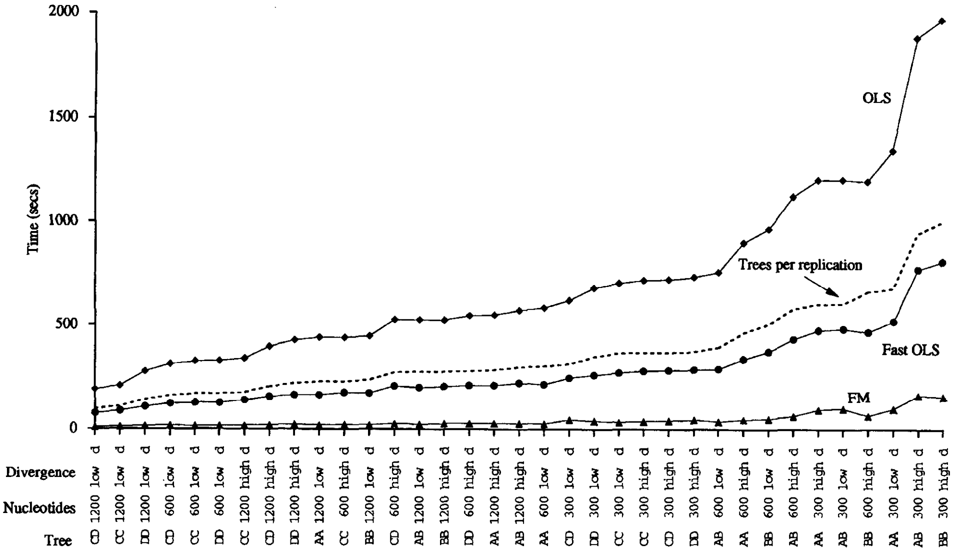

FIG. 4.—Comparison of the time required to complete search for the ME tree in 200 simulation replications for 12-taxon model trees (fig. 5) by three different methods of computing S values based on the estimates of branch lengths by ordinary least squares (OLS), optimized OLS (FastOLS), and Fitch and Margoliash (FM) methods. On abscissa, the name of the tree from figure 5, the number of nucleotides, and the evolutionary divergence considered in the simulation are shown. The number of trees examined per replication is shown with dotted lines. All results are from proportional search with a 10% search factor; simulations were conducted on an IBM PC 486/66 MHz computer.

licate data sets in each case of simulation, unless otherwise mentioned.

Model Trees and Realized Trees

In the study of reconstruction of phylogenetic trees by computer simulation, it is important to distinguish between the model tree and the realized tree. The model tree is the tree with all branches expressed in terms of expected number of nucleotide substitutions per site, whereas the realized tree is the one with branch lengths equal to the actual number of substitutions per site (Nei 1987). This difference arises because nucleotide substitutions occur stochastically, and, thus, the realized number of substitutions for a branch in a computer simulation may be different from the expected number of substitutions per sequence when the expected number of substitutions per site (x) multiplied by the number of sites (n) is small.

For example, in the model tree A in figure 5, $x{\cdot}n = 0.00625 \times 300 = 1.8$ for the case of small divergence and the small number of nucleotides. Since the number of substitutions per sequence (N) follows the Poisson distribution with mean and variance equal to 1.8, it varies substantially among different replications of the simulation. Furthermore, N will be 0 with a probability of $e^{-1.8} = 0.165$. In tree A, there are three such short interior branches. Therefore, the probability that at least one of the three interior branches has $N = 0$ is $1 - (1 - 0.165)^3 = 0.418$. This indicates that with a probability of 0.418, the realized tree is expected to be a multifurcating tree and its topology is different from that of the model tree (table 3; $P_{R=M}$).

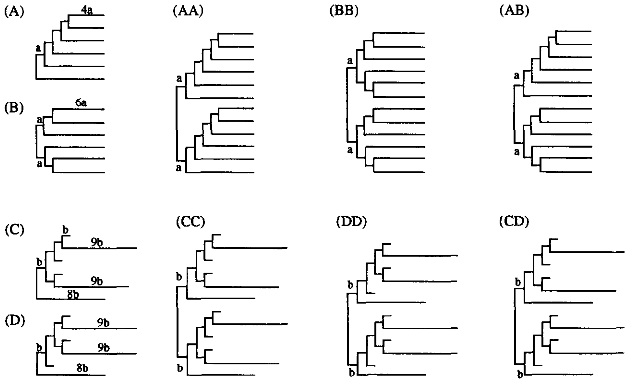

FIG. 5.—The model trees used for simulation. Each interior branch is one unit long (a for constant and b for variable-rate trees) and the length of external branches are given in multiples of a or b. Low divergence refers to $a = 0.00625$, $b = 0.01$ (maximum pairwise divergence of ~0.1 substitutions per site) and high divergence refers to $a = 0.0625$, $b = 0.05$ (maximum pairwise divergence of ~1.0 substitutions per site).

Table 3
Efficiency of the NJ Method in Recovering Model and Realized Trees

	300 NUCLEOTIDES			600 NUCLEOTIDES		
MODEL	$P_{R=M}$	$P_{NJ=M}$	$P_{NJ=R}$	$P_{R=M}$	$P_{NJ=M}$	$P_{NJ=R}$
A	59	56	78	93	82	87
B	70	61	80	95	82	85
C	87	75	81	99	94	95
D	84	73	83	99	94	95
AA . . .	23	20	57	82	66	72
BB	2	19	53	86	59	67
AB . . .	27	20	52	83	60	69
CC	64	49	65	97	85	88
DD . . .	66	50	66	98	90	91
CD . . .	64	48	63	97	87	88

NOTE.—Proportion of replications in which the topology of the realized tree is the same as that of the model tree ($P_{R=M}$), the topology of the NJ tree is same as that of the model tree ($P_{NJ=M}$), and the topology of the NJ tree is same as that of the realized tree ($P_{NJ=R}$), for the case of low divergence (~0.1 substitutions per site).

Table 4
Efficiency of Identifying the Correct Leading Taxa in Every Step of Taxon Pairing with 300, 600, and 1,200 Nucleotides

	P_I (P_I')			P_{II} (P_{II}')			P_{III} (P_{III}')		
MODEL	300	600	1,200	300	600	1,200	300	600	1,200
Low divergence:									
A ...	99 (100)	100	100	93 (99)	99	100	56 (78)	82 (87)	96
B ...	99 (100)	100	100	96 (89)	99 (96)	100	61 (81)	82 (85)	96
C ...	100	100	100	99 (100)	100	100	75 (81)	94 (95)	99
D ...	100	100	100	98 (99)	99 (100)	100	73 (93)	94 (95)	99
AA ..	97 (99)	99 (100)	100	70 (95)	95 (98)	100	20 (57)	66 (72)	93
BB ..	96 (98)	100	100	69 (94)	96 (97)	99	19 (54)	60 (67)	93
AB ..	96 (99)	99	100	72 (96)	92 (96)	100	20 (52)	60 (69)	92
CC ..	99	100	100	91 (96)	99	100	49 (65)	85 (88)	99
DD ..	99	100	100	91 (96)	98 (99)	99	50 (66)	90 (91)	99
CD ..	99	100	100	88 (95)	99	100	48 (63)	87 (88)	98
High divergence:									
A ...	99	100	100	89	96	100	45	66	95
B ...	99	100	100	93	98	100	44	71	93
C ...	99	100	100	98	100	100	65	88	98
D ...	100	100	100	98	100	100	66	88	99
AA ..	93 (98)[a]	99	100	71 (89)[a]	93	99	14 (28)[a]	47	90
BB ..	96 (98)[a]	99	100	67 (76)[a]	91	99	9 (14)	39	87
AB ..	92 (98)[a]	99	100	66 (81)[a]	91	100	11 (20)[a]	41	90
CC ..	99	100	100	94	98	100	48	81	97
DD ..	99	100	100	94	99	100	46	85	97
CD ..	98	100	100	93	98	100	47	76	99

NOTE.—P_I: proportion of replicates in which the selected pair contained at least one leading taxon; P_{II}: correct leading taxon was selected; P_{III}: the taxa in the selected pair were each other's neighbors. Values in the parentheses were obtained by using the realized tree as the correct tree (P_I', P_{II}', P_{III}'; shown if different).

[a] With *p* distance.

Most tree-building methods are intended to estimate realized trees rather than the model trees. Therefore, it is important to compare the topology of a reconstructed tree with that of the realized tree. If we compare the reconstructed tree with the model tree, we may underestimate the efficiency of a tree-building method in recovering the correct tree. Table 3 shows the results of comparison of the reconstructed tree by the NJ method with the model tree and realized tree for the case of $n = 300$. It is clear that the reconstructed tree agrees with the realized tree more often than with the model tree. For this reason, we will also consider the comparison of reconstructed trees with realized trees in this paper. Of course, this problem becomes trivial when $x \cdot n$ is large.

Efficiency of Finding Leading Taxa

In the new algorithm, the identification of correct leading taxa in different steps of taxon pairing is critical, because otherwise the correct tree will not be included in the tree space searched. Since a leading taxon is selected from the pair of taxa with smallest S_{ij} in each step, the efficiency of finding a correct leading taxon depends directly on the likelihood that the selected pair contains at least one taxon that has a neighbor in the correct tree (P_I) and that the correct taxon is chosen as a leading taxon from this pair (P_{II}). Table 4 shows how often these conditions are met in every step of taxon pairing. Since realized trees may contain multifurcations in some cases (table 3), the values of P_I and P_{II} for the case when the realized trees were used for comparison (P_I' and P_{II}', respectively) are also shown in this table.

Table 4 shows that the selected pair almost always contained at least one taxon that had a potential neighbor in the correct tree (a leading taxon), irrespective of the sequence length and the magnitude of evolutionary divergence studied. The approach adopted for choosing a leading taxon from the selected pair also appears to be efficient, except for the case of small number of nucleotides and low divergences where the realized tree often differ contain zero-length branches ($P_I' > P_I$ and $P_{II}' > P_{II}$). Selection of correct leading taxa also appears to be difficult for constant rate trees *AA, BB,* and *AB* (fig. 5) for the case of small number of nucleotides and high divergence. However, the use of proportion of nucleotide differences (*p* distance), instead of the Jukes-Cantor (1969) distance (*d*), improves the values of P_I and P_{II} significantly, even though the *p* distance is not an unbiased estimate of the evolutionary divergence (see also Rzhetsky and Nei 1993). It appears that the presence of many large pairwise distances ($d \geq 1$) whose estimates have rather large variances, whenever the sequences are short, affects the ability of the new algorithm to select correct leading taxon. For instance, in the model tree given in figure 5*AA*, the variance of the largest expected *d* (=1.125 substitutions per sites) is about 20 times that of the variance of the corresponding *p* distance (=0.582). For the case of $n = 300$, the estimate of $d = 1.125$ has a standard error of ±0.127. Thus, $d = 1.0$ (second largest expected distance in tree *AA*) lies

Table 5
Efficiency of the ME Method in Recovering Model and Realized Trees

	Low Divergence			High Divergence		
Tree	300	600	1,200	300	600	1,200
A	56 (78)	83 (87)	96	45	65	95
B	61 (81)	82 (85)	96	44	72	93
C	74 (81)	94 (95)	99	66	88	98
D	73 (83)	94 (95)	99	66	88	99
AA ...	20 (56)	65 (72)	93	15	42	89
BB ...	20 (54)	59 (67)	93	9	38	86
AB ...	20 (53)	60 (68)	92	11	38	89
CC ...	49 (63)	86 (88)	99	48	81	97
DD ...	52 (68)	92 (92)	99	53	89	98
CD ...	50 (65)	87 (89)	98	52	79	98

Note.—Proportion of replicates where the ME tree was identical with realized tree are shown in the parenthesis (if different). Saitou and Nei's method (500 replications) and the OLS method (200 replications) gave almost identical results.

within one standard error of $d = 1.125$, and these similar magnitudes of pairwise distances with large sampling errors may obscure the phylogenetic information in the present case. At any rate, since the S value of the correct tree is expected to be the smallest, P_{II} (and P_{II}') represents the upper bound on the efficiency of the new algorithm in recovering the correct tree in our simulations, which are quite high.

The proportion of replications in which the taxa in the selected pair are each other's neighbors in every step of taxon addition (P_{III} and P_{III}') are also given in table 4. P_{III} is also the efficiency of the NJ tree in recovering the true tree. Clearly, the NJ method is not expected to perform very well with a large number of sequences (e.g., 12 in the present case) because both taxa in the selected pair are rarely neighbors if sequence lengths are short or only moderately long. P_{II} and P_{II}' are substantially larger than P_{III} and P_{III}', respectively, in most cases.

Efficiency of the ME Method in Finding the Correct Tree

The efficiency of the ME method in recovering the correct tree is similar to that of the NJ method (tables 3 and 5), and, in both cases, as the number of taxa in the model tree increases the probability of recovering the correct tree decreases. Furthermore, the efficiency of the ME method appears to be slightly lower for high divergence as compared to the low divergence cases. For model trees consisting of six taxa (fig. 5*A–D*), these efficiencies are identical to those reported by Saitou and Imanishi (1989) and by Rzhetsky and Nei (1992), who used the same model trees in their simulations.

Although the correct tree is included in the tree space searched using the new algorithm (table 4; P_{II}), the probability that the ME tree is the correct tree is considerably lower (table 5). Thus, the correct tree often does not have the smallest S value. This may be due to sampling errors associated with the estimation of pairwise distances and the presence of zero-length interior branches in the realized tree. In both these scenarios, we expect to find many bifurcating topologies for which the S values are quite similar (or statistically equal). To examine if this is the case, we determined the percent difference in the S values of the NJ tree (S_{NJ}) and the model tree (S_M) whenever the model tree did not have the smallest S value. We also counted the number of trees examined in the search for which the difference in their S values from the S_{NJ} was less than or equal to the difference between the S values of the NJ tree and the model trees, i.e., $S - S_{NJ} \leq S_M - S_{NJ}$ or $S < S_{NJ}$. The NJ tree was used as a reference because it can always be obtained and because its S value is expected to be close to that of the ME tree. For the case of 300 nucleotides and low divergence, we find that the model tree was usually present within 1%–2% (S value) neighborhood of the NJ tree (fig. 6*A*); this much difference in S is generally not statistically significant (e.g., by Rzhetsky and Nei 1992 test). This also holds for the high-divergence case for sequences of length 300. With 600 nucleotides (fig. 6*B*), the model tree is found within 1% neighborhood of S_{NJ} for the low- as well as the high-divergence case. For longer sequences, we found that the NJ, the ME, and the correct trees were almost always identical in topology.

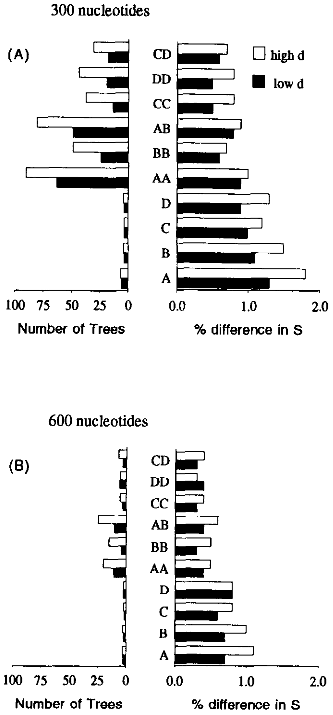

Fig. 6.—The largest percent difference between S_M and S_{NJ} that is needed to include model trees in the set of equally likely trees in >90% replications, whenever the model tree was examined in the search but was different from the NJ tree. Mean number of trees for which $S - S_{NJ} \leq S_M - S_{NJ}$ or $S < S_{NJ}$ are also shown.

Clearly, the mean number of trees examined for which $S - S_{NJ} \leq S_M - S_{NJ}$ (or $S < S_{NJ}$) is the largest for the case of shortest sequence lengths and high divergence (fig. 6). As expected, these numbers are smaller for the case of 600 nucleotides as compared to that of 300 nucleotides. This is because of fewer interior branches with zero-length in the realized trees (see table

3) and because the pairwise distances are estimated with smaller variances as the sequence length increases or when the pairwise distances are small. For longer sequences, these errors are further reduced, and the topology of the ME tree was almost always identical with that of the model tree.

For six-taxon trees in figure 5*A–D,* Rzhetsky and Nei's (1992) results showed that the correct tree was almost always present within two topological rearrangements from the NJ tree and that the average topological distance (d_T; Robinson and Foulds 1981) was about 2.3. Therefore, search for the ME tree in $d_T \leq 4$ neighborhood of the NJ tree is likely to contain the correct tree most of the time. However, we find that the average d_T between the NJ tree and the correct tree is more than 4 in the case of 300 nucleotides for 12-taxa trees (results not shown). Therefore, $d_T > 4$ neighborhood of the NJ tree is also needed to be searched if Rzhetsky and Nei's (1992) method is used. With a larger number of taxa and/or slower evolving sequences, the average d_T between the NJ and the correct tree is likely to increase further. In such cases, the search for the ME tree using the present algorithm or improved methods suggested by Rzhetsky and Nei (1994) would be more appropriate.

Computational Efficiency of the New Algorithm

Computationally Efficient Alternatives to the Full Search

In simulations with model trees *A–D* (fig. 5), we examined all possible potential neighbors of the leading taxa in each step of taxon pairing. We then repeated these simulations by employing the proportional search factors for restricting the tree space searched. The results obtained in this way were almost identical with those obtained with the full search, but the numbers of trees examined per replication (and the time taken) were up to two times smaller when the proportional search factor of 10% was used (results not shown).

In simulations with 12-taxa model trees, the full search was not computationally feasible; it required the examination of about 20 million trees in each of the 6·2·3·500 simulation replications (model trees × rates × lengths × replications). Instead, we used search factors of 10% and 20% in the simulation study. The results obtained from these experiments were almost identical. However, the number of trees examined per replication in the case of 20% search factor was 10–30 times larger than those in the 10% search. As expected, the number of trees examined in the search decreased with increasing sequence length (fig. 4). A large number of topologies were examined when 10% search factor was used for constant-rate trees *AA, BB,* and *AB* with a small number of nucleotides and high divergence. This is because many topologies exist that are statistically indistinguishable from the ME tree in their *S* values (fig. 6*A*).

Use of a Simple Method for Estimating Branch Lengths

As mentioned earlier, the computation of branch lengths by the Fitch and Margoliash (1967) method is

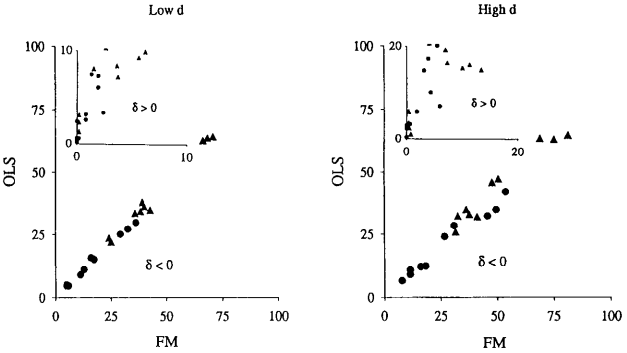

FIG. 7.—The proportions of *all* replicates in which the *S* value for NJ tree (S_{NJ}) was *smaller* than that for the model tree (S_M; $\delta < 0$), where *S* values were computed from branch lengths estimated by the FM and OLS methods. The proportions of *all* replications in which the NJ tree had larger *S* value than the model tree ($\delta > 0$) are also shown for each case. ▲ = 300 nucleotides, ● = 600 nucleotides.

much faster than that by the OLS method. However, the *S* values for a tree obtained using the branch lengths estimated by FM and OLS methods may be different and, therefore, the ME tree found using these two methods of estimating *S* values may not be identical in topology. This may adversely affect the efficiency of the ME method in recovering the correct tree. Clearly, the ME criterion based on the FM estimates of *S* values is expected to perform with the same efficiency as the OLS method only if the sign of the difference of *S* values for any given tree and the correct tree is the same for the FM and OLS methods. In this case, the use of either method would result in the selection of the same tree as the ME tree. To study this problem, we used the NJ tree as a reference, because the ME tree is not known beforehand, and computed the proportion of *all* replicates in which the estimate of *S* computed by OLS and FM methods were smaller for the NJ tree than for the model tree ($\delta < 0$) whenever the NJ and model trees were not identical. The NJ tree was used as a reference because its *S* value is generally very close to that of the ME tree, which would enable us to detect even small differences in efficiencies of FM and OLS methods. The results from FM and OLS estimation methods were highly correlated for the cases of low as well as high divergence (fig. 7). Furthermore, the efficiency of the ME method with the FM estimates was almost identical to that with the OLS estimates, as given in table 5. These observations are consistent with the fact that the NJ and ME methods of phylogenetic reconstruction have similar theoretical bases (Saitou and Nei 1987; Rzhetsky and Nei 1992; Gascuel 1994). This observation and the fact that the use of the FM method requires only a fraction of the time needed in the OLS method suggest that the FM method of estimating branch lengths may be useful when searching for the ME tree for data sets containing a large number of taxa (>20).

Discussion

In this paper, we have presented an algorithm that searches for an optimal tree heuristically by minimizing the sum of branch lengths, which can be computed by ordinary least squares or by the Fitch and Margoliash method. Using computer simulations, we have shown that the tree space searched almost always contains the correct tree, unless the number of nucleotides considered is rather small (table 4; P_{II}). This holds true for both the "full-search" and the proportional-search strategies. Furthermore, the performance of the ME method in our simulations with proportional-search factor for six-taxon trees is almost identical to that obtained by Rzhetsky and Nei (1992) in their topological distance method. However, unlike in their method, the sum of branch lengths at different stages of taxon pairing, not the difference in topology, is used as the primary criterion for generating alternative topologies in our algorithm. Thus, the number of trees examined by the stepwise search algorithm is influenced by the number of taxa in the study, stochastic errors of distance estimates, and the instability of the tree. For instance, in a stepwise search with a 10% search factor, the largest number of trees was examined for the cases in which the efficiency of the ME (and NJ) method was lowest in the 12-taxon trees (see fig. 4 and table 5). In contrast, the number of trees with $d_T = 2$ (or 4) would be the same (or similar in magnitude) as long as the number of taxa in the data set is equal in the topological distance method (see, however, Rzhetsky and Nei 1994). In the "bootstrap" method (Rzhetsky and Nei 1994), the number of trees examined will depend directly on the number of replications conducted.

The stepwise nature of the new algorithm allows us to reject many partial trees (and thus search paths) that would inevitably lead to trees with larger S than the temporary ME tree. This optimization and the reduction of the computational time overhead attained by using FastOLS or FM methods for estimating S values make the search for the ME tree practical for large data sets. In fact, some of these optimizations can easily be incorporated into the Rzhetsky and Nei (1992, 1994) methods to improve their efficiency. At this moment, a tentative comparison of the computational efficiency of the Rzhetsky and Nei (1994) methods with that of the new algorithm would be similar to the comparison of OLS and FastOLS (or FM) curves in figure 4, if we assume that both methods examined equal number of trees.

As shown in table 5, the probability that the smallest tree found is identical to the correct tree in topology is rather low even for 12-taxon trees when the evolutionary distances are estimated with large sampling errors. This is also the case for the NJ method and may be true for other tree-making methods whose performance usually is equal to or worse than that of the NJ method (Nei 1991). In an empirical study of the relative efficiency of different tree-making methods, it was found that the performance of NJ and ME methods were similar in reconstructing a known tree of 11 vertebrate species for different mitochondrial protein-coding genes (Russo, Takezaki, and Nei 1996). In addition, the topological distances between the correct tree and the ME and NJ trees were also almost equal in their study. Thus, our simulation results agree closely with those in Russo, Takezaki, and Nei's empirical study.

Figure 6 shows that whenever the NJ tree (or ME tree) is not the correct tree, the S value of the correct tree is only 1%–2% different from that of the NJ tree (fig. 6; table 5 in Rzhetsky and Nei 1992) and that many trees with similar S values exist. Thus, the new algorithm can be used with a 10% or 20% search factor to identify most (or all) of the trees that are plausible under the ME criterion. These trees can then be compared with other trees in the set or some given tree by testing the statistical significance of the difference in their S values using the OLS method (Rzhetsky and Nei 1992, 1994). It is also possible to build a confidence set of trees (Navidi, Churchil, and von Haeseler 1991) such that the difference in S values between the trees included in this set and the minimum tree (or the NJ tree) is not more than a given cut-off value (specified as absolute difference or in units of standard deviations). For data sets with many sequences, we could identify clusters of sequences that form monophyletic groups in all the equally good trees and then conduct four-cluster analysis (Rzhetsky, Kumar, and Nei 1995) to ascertain a preferred branching order.

We have explored the efficiency of our algorithm using simulations with only a few model trees. In some limited simulations with model trees consisting of 24 sequences, we found results similar to the ones presented above. In the future, we plan to conduct extensive simulation analyses to further evaluate the usefulness of the new algorithm.

Acknowledgments

I thank Drs. Andrey Rzhetsky and Ziheng Yang for valuable suggestions on an earlier draft of this manuscript, and Dr. Masatoshi Nei for motivating this work. This research was supported by grants from the National Institute of Health and the National Science Foundation to M.N. A simple DOS-based computer program is available on request from the author. In the future, this method will be made available in a more user-friendly computational environment in the MEGA program package (Kumar, Tamura, and Nei 1993).

Appendix: Selecting the Leading Taxon

If S_{ab} is the smallest S value in a given step of taxon pairing, there are four different possibilities regarding the neighbor status of a and b in the "true" tree. (i) a and b are immediate neighbors (e.g., 1 and 2 in fig. 1), (ii) neither a nor b has a neighbor (e.g., 3 and 4), (iii) both a and b have neighbors, but they are not each other's neighbors (e.g., 1 and 6), (iv) a has a neighbor, b does not (e.g., 1 and 3) and vice versa. In case (i) choice of a leading taxon is trivial, whereas in case (ii) the choice of any taxon from the pair would result in an error. We ignore the possibility that neither a nor b has a neighbor, an assumption that does not seem to affect the results seriously (table 4; P_I). Now, either a or b has a neighbor. If a has a neighbor u in the correct tree, we would expect S_{ua} to be the smallest in the set $\{S_{ia}, S_{jb};$

where $i, j \neq a, b\}$ (Saitou and Nei 1987). On the other hand, if b has a neighbor v, S_{vb} is expected to be the smallest in the set $\{S_{ia}, S_{jb}$; where $i, j \neq a, b\}$. Therefore, to select a leading taxon from the pair (a, b), we find the smallest S_{ij} from the set $\{S_{ia}, S_{jb}$; where $i, j \neq a, b\}$. If the minimum S_{ij} is found in the set (S_{ia}'s), it indicates that a is the leading taxon, otherwise b is chosen. In the following, we present a simple application of the above argument (not a proof) for four- and five-taxon cases.

Four-taxon Case

Because every taxon in a tree of four taxa has a true neighbor, the leading taxon chosen also has a true neighbor. Therefore, the new algorithm will always find the ME tree.

Five-taxon Case

In a bifurcating tree consisting of five taxa, there is only one unlabeled tree (fig. 3*A*). In the first step of search, the smallest S_{ij} can be either from the set $\mathbf{X} = \{S_{12}, S_{45}, S_{14}, S_{15}, S_{24}, S_{25}\}$ or set $\mathbf{Y} = \{S_{13}, S_{23}, S_{34}, S_{35}\}$. If the smallest S_{ij} (S_{ab}) is from set $\mathbf{X}$, either taxon can be chosen as a leading taxon since both taxa in every pair have neighbors. However, if the S_{ab} is from the set $\mathbf{Y}$, in which taxon 3 does not have a neighbor, we examine S_{ia} $(i \neq a, b)$ and S_{jb} $(j \neq a, b)$. Let us assume that S_{13} is the smallest S_{ij}. We find the minimum S from the set of S_{ij} values $\{S_{12}, S_{14}, S_{15}, S_{23}, S_{34}, S_{35}\}$. Since the two branches that are connected to the interior node where the branch leading to taxon 3 joins are nonzero, S_{12} is expected to be the smallest of all the S values in this set (Saitou and Nei 1987). In this case, taxon 1 is common to both S_{13} and S_{12}, and is, therefore, correctly chosen as the leading taxon. This argument applies to other pairs in $\mathbf{Y}$ because they are equivalent in terms of their position on the tree in figure 3*A*. Obviously, if we combine two taxa correctly in a five-taxon tree, we are left with a four-taxon tree for which the minimum tree is guaranteed to be found with the present algorithm.

Similar argument can be applied to a six- or higher taxon tree but the explanation may be more complicated depending on the number of unlabeled trees for the given set of taxa.

LITERATURE CITED

FITCH, W. M., and E. MARGOLIASH. 1967. Construction of phylogenetic trees. Science **155**:279–284.

GASCUEL, O. 1994. A note on Sattath and Tversky's, Saitou and Nei's, and Studier and Keppler's algorithms for inferring phylogenies from evolutionary distances. Mol. Biol. Evol. **11**:961–963.

JUKES, T. H., and C. R. CANTOR. 1969. Evolution of protein molecules. Pp. 21–132 *in* H. N. MUNRO, ed. Mammalian protein metabolism. Academic Press, New York.

KIDD, K. K., and L. A. SGARAMELLA-ZONTA. 1971. Phylogenetic analysis: concepts and methods. Am. J. Hum. Genet. **23**:235–252.

KUMAR, S., K. TAMURA, and M. NEI. 1993. MEGA: Molecular Evolutionary Genetics Analysis. Institute of Molecular Evolutionary Genetics, University Park, Penn.

NAVIDI, W. C., G. A. CHURCHIL, and A. VON HAESELER. 1991. Methods for inferring phylogenies from nucleic acid sequence data by using maximum likelihood and linear invariant. Mol. Biol. Evol. **8**:128–143.

NEI, M. 1987. Molecular evolutionary genetics. Cambridge University Press, New York.

NEI, M. 1990. Molecular evolutionary genetics. Japanese edition (revised). Baifukan, Tokyo, Japan.

NEI, M. 1991. Relative efficiencies of different tree making methods for molecular data. Pp. 133–147 *in* M. M. MIYAMOTO and J. L. CRACRAFT, eds. Recent advances in phylogenetic studies of DNA sequences. Oxford University Press, Oxford.

ROBINSON, D. F., and L. R. FOULDS. 1981. Comparison of phylogenetic trees. Math. Biosci. **53**:131–147.

RUSSO, C. A. M., N. TAKEZAKI, and M. NEI. 1996. Efficiencies of different genes and different tree-building methods in recovering a known vertebrate phylogeny. Mol. Biol. Evol. (in press).

RZHETSKY, A., S. KUMAR, and M. NEI. 1995. Four-cluster analysis: a simple method to test phylogenetic hypotheses. Mol. Biol. Evol. **12**:163–167.

RZHETSKY, A., and M. NEI. 1992. A simple method for estimating and testing minimum-evolution trees. Mol. Biol. Evol. **9**:945–967.

RZHETSKY, A., and M. NEI. 1993. Theoretical foundation of the minimum-evolution method of phylogenetic inference. Mol. Biol. Evol. **10**:1073–1095.

RZHETSKY, A., and M. NEI. 1994. METREE: a program package for inferring and testing minimum-evolution trees. Comput. Appl. Biosci. **10**:409–412.

SAITOU, N., and M. IMANISHI. 1989. Relative efficiencies of the Fitch-Margoliash, maximum-parsimony, maximum-likelihood, minimum-evolution, and neighbor-joining methods of phylogenetic reconstructions in obtaining the correct tree. Mol. Biol. Evol. **6**:514–525.

SAITOU, N., and M. NEI. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**:406–425.

STUDIER, J. A., and K. J. KEPPLER. 1988. A note on the neighbor-joining algorithm of Saitou and Nei. Mol. Biol. Evol. **5**:729–731.

TAKASHI GOJOBORI, reviewing editor

Accepted January 2, 1996